

The opinion in support of the decision being entered today was *not* written for publication and is *not* binding precedent of the Board.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte MICHAEL D. STAPF

Appeal No. 2006-0888
Application 09/291,535¹

ON BRIEF

MAILED

JUN 30 2006

PAT. & T.M. OFFICE
BOARD OF PATENT APPEALS
AND INTERFERENCES

Before BARRETT, MEDLEY, and MOORE, *Administrative Patent Judges*.

BARRETT, *Administrative Patent Judge*.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134(a) from the final rejection of claims 1-26.

We reverse.

BACKGROUND

The invention relates to an interface which provides data to an Enterprise Resource Planning (ERP) program. ERP systems are integrated information systems that

¹ Application for patent filed April 14, 1999, entitled "Interface for an Enterprise Resource Planning Program."

have been developed to serve a variety of departments within an enterprise, e.g., a corporation. ERP systems typically include software that manages information for use in manufacturing, order entry, accounts receivable and payable, general ledger, purchasing, warehousing, transportation, and human resources. ERP systems are commercially available from companies such as SAP.

One problem with ERP systems is loading or interfacing the ERP systems with data from existing systems, so-called "legacy data." Conventionally, legacy data is provided to the ERP system through a custom-built interface. The steps 1-11 at pages 1-2 of the specification are said to be typically used to provide legacy data to the ERP system. This process is time consuming and requires the expertise of a number of different people and is subject to misunderstanding at each point in the process.

The invention uses a "parameter file" that maps data in a legacy data file with the appropriate screens, fields, and navigation of an ERP system, which can be created using any acceptable text editor and does not require knowledge of a unique programming language. The format of a parameter file is shown in Fig. 2, specification, page 6, and a parameter file example is described at pages 15-16.

Claim 1 is reproduced below.

1. A method for interfacing with an enterprise resource planning system, the method comprising:

providing a file containing data to be loaded into the enterprise planning system (the "data file");

creating a file containing at least one parameter (the "parameter file"), wherein the parameter file maps the data from the data file to screens of the enterprise resource planning system; and

processing each record in the data file according to the parameters in the parameter file to execute screens of the enterprise resource planning system so as to provide the data from the data file to the enterprise resource planning system.

THE REFERENCES

The Examiner relies on the following references:

Glowny	5,805,897	Sep. 8, 1998
Geller	5,844,554	Dec. 1, 1998
Mazzario	5,084,815	Jan. 28, 1992

THE REJECTIONS

Claims 1-10, 13, 14, 16-21, and 23-26 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Geller and Official Notice that processing each record in a file was well known. In the Supplemental Examiner's Answer, the Examiner provided Mazzario in support of this finding of Official Notice.

Claims 11, 12, 15, and 22 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Geller and Glowny.

We refer to the Final Rejection, the Examiner's Answer, and the Supplemental Examiner's Answer for a statement of the examiner's rejection, and to the Brief, the Reply Brief, and the Supplemental Reply Brief for a statement of Appellant's arguments thereagainst.

DISCUSSION

The invention

The steps 1-11 at pages 1-2 of the specification are said to be typically used to provide legacy data to the ERP system. The specification does not clearly describe what steps are avoided by the invention. It appears that all steps are required in appellant's invention except for the step of writing a program in "ABAP" (Advanced Business Application Programming language by SAP) code; i.e., someone must determine where the data goes in the SAP (step 1), must manually execute the screens to determine what buttons are pushed to navigate (step 2), must define a fixed-format file for the legacy data (step 3), must write a program to extract the legacy data from the legacy system (step 4) (i.e., convert a "record" to a "flat" file), must write an SAP Batch Data Conversion (BDC) program to read the flat file extracted from the legacy system (step 5), and must test and execute the program (steps 6-11). Stated differently, someone must define the source structure (the structure of data in the source file); define the target structure (the structure of SAP that receives data); define the field mapping (the mapping between the source and target structure, with conversions, if any); specify the location of the source file; read data

(legacy data in spreadsheet tables and/or sequential files); batch convert data (from the source format into the target format); import data (to the database used by the SAP R/3 application); and perform testing and verification. Instead of the mapping being "hard coded" into an ABAP program, appellant describes the mapping is stored in a parameter file. If we are wrong in our understanding, appellant should request rehearing.

The specification does not describe what kind of program uses the parameter file. If a conventional program, such as SAP, is used, then the question is whether a parameter file is a conventional feature. Based on our search on the Internet, those skilled in the art *today* are aware of several ways to upload legacy files to SAP:

- Build a program in ABAP.
- Use an SAP tool called LSMW (Legacy Systems Migration Workbench).
- Use an internal test tool called CATT (Computer Aided Test Tool).
- Use some third party tool such as TxShuttle from Winshuttle.
- For some business objects, use the SAP mass maintenance transactions such as
 - - MASS or MM17.
- If SAPGui scripting is enabled, the user can "script" an upload/modify job by looping through say an Excel table with vbscript.

It is not known what methods were known at the time of the invention, but apparently ABAP and LSMW were known. LSMW is an SAP R/3 based tool that supports the one-time or periodic transfer of data from non-SAP systems ("legacy systems") to SAP

systems. LSMW helps reduce the cost and time it takes to migrate legacy data by allowing transfer data from a variety of sources without any programming rather than having to do the custom programming usually needed for legacy migration. We have no experience in SAP and LSMW or what files are created and used for those programs (e.g., whether LSMW creates a parameter file). We presume appellant would have disclosed if the invention used a conventional program or tool such as SAP's LSMW to implement the invention, since that would be the closest prior art.

Obviousness

Content of Geller

Geller relates to a method for creating a user interface and handling constraints in a product configurator computer program. A new type of sales automation software, known as "configuration" software or "configurators," has recently emerged to assist in the product sales process (col. 1, lines 24-26). With a configurator, a salesperson with a computer engages in an information gathering session with a prospective customer, and receives information about the customer's product needs, such as budget constraints, model preferences, features desired, configuration options, etc. (col. 1, lines 29-34). This information is interactively entered into the configuration software, and the software responds by providing indications that certain configurations are not valid, computing the price of the product as configured, generating a purchase order, and so on (col. 1, lines 34-41). Prior art configurators traditionally used an interpreted-engine based

architecture, which has several drawbacks (col. 1, line 49, to col. 2, line 7), and has been typically "hard coded" so that only a programmer was able to effect changes, and changes were difficult and time consuming to make (col. 2, lines 8-19).

Geller discloses a product configurator computer program that allows a sales engineering force--who need not be computer programmers--to create and update a product configurator computer program (col. 2, lines 20-23). An integrated user interface builder developer allows the configuration developer to create both a visual user interface (for the salesperson) and the underlying configuration logic (col. 3, lines 24-27). The process for creating and using configuration software is a six-step process as shown in Fig. 2 (col. 8, line 1, to col. 9, line 28). The developer environment allows creation and editing of parameters and constraints, accessing existing data of the enterprise, and compiling an executable configuration software program (col. 8, lines 4-9). A parameter corresponds to a data object variable or field, which is used to contain, save, calculate, compare and display information (col. 10, lines 64-66; col. 18, line 26, to col. 19, line 16); e.g., a parameter "Color" may have a value "Blue" (col. 18, lines 43-48). Constraints are used to establish rules that define the valid relationships between parameters (col. 11, lines 3-11). Existing data of the enterprise are stored in an existing ERP database (col. 8, lines 9-12). The executable configuration software is operative to execute SQL queries on any ERP data (col. 8, lines 51-56).

Arguments and responses

Appellant argues that Geller relates to the accessing of data stored in an Enterprise Resource Planning (ERP) database, *not* to providing data to the ERP system (Brief, p. 10). The examiner finds that Geller provides data into the ERP system because the configuration software executes SQL queries on ERP data (Supp. Exm'r Answer, p. 10).

Appellant argues that the parameters of Geller relate to product searches in the ERP, *not* to mapping data into the ERP as disclosed and claimed (Brief, pp. 10-11). The examiner responds that Geller teaches (at col. 4, lines 40-46) storing parameters associated with the product configurator in a hierarchical arrangement, which the examiner interprets to mean that the parameters map the data of the data file (Supp. Exm'r Answer, p. 11).

Appellant argues that Geller does not teach a file containing data to be loaded into the ERP system, but only accesses data already in an ERP database (Br11-12). The examiner repeats that Geller provides data into the ERP system because the configuration software executes SQL queries on ERP data (Supp. Exm'r Answer, p. 11).

Analysis

Claim 1 is representative.


It is immediately apparent that Geller does not convert data from a legacy data file to ERP screens so as to provide data from the data file to the ERP systems, as disclosed. Geller has an ERP database 21, but this corresponds to the data of the ERP system.


While Geller does allow a non-programmer to implement the configuration software, which is somewhat analogous, in general theory, to appellant's disclosed method of allowing a non-programmer to convert legacy data to an ERP format, Geller does not meet the claim limitations. We have tried to understand how the examiner might have broadly interpreted the claims to be unpatentable over Geller, but find that the claims recite limitations not found or suggested in Geller.


The limitation of "providing a file containing data to be loaded into the enterprise planning system (the 'data file')," standing alone, might be broadly interpreted to be met by the ERP database 21 in Geller. However, the "parameters" in Geller correspond to data variables, not something that "maps data from the data file to screens of the enterprise resource planning system." The examiner's reasoning that the parameters map data does not explain how the parameters map data from a data file "to screens of the enterprise resource planning system," as claimed. The mere fact that data representing a parameter (e.g., "Blue" for the parameter "Color") is displayed on a screen does not mean that the parameter maps data from a data file to a screen of the ERP. Geller does not describe anything that could be construed as batch conversion of records in a data file to the enterprise resource planning system, which is found in the limitation of "processing each record in the data file . . . so as to provide the data from the data file to the enterprise resource planning system." While the examiner interprets appellant's mention of this limitation (at Br12) to contest the finding that processing each record was well known in

the art (SEA5-6), appellant is only arguing that there is no teaching of converting a record from a data file to the ERP, much less converting each record in the data file. Mazzario, which has been applied to support the finding of Official Notice that converting each record in a data file was known, does not cure the deficiencies of Geller with respect to the rejection of claim 1. Glowny, which is applied to dependent claims 11, 12, 15, and 22, also does not cure the deficiencies of Geller with respect to the rejection of the independent claims. We conclude that the examiner has failed to establish a prima facie case of obviousness. The rejections of claims 1-26 are reversed.

REVERSED


LEE E. BARRETT
Administrative Patent Judge


SALLY C. MEDLEY
Administrative Patent Judge


JAMES T. MOORE
Administrative Patent Judge

)
)
)
)
)
) BOARD OF PATENT
) APPEALS
) AND
) INTERFERENCES
)
)
)
)
)

Appeal No. 2006-0888
Application 09/291,535

FOGG AND ASSOCIATES, LLC
P.O. BOX 581339
MINNEAPOLIS, MN 55458-1339